Towards Using Speech Melody to Guide Large Language Models

David Porteš

Faculty of Informatics,
Masaryk University,
Botanická 68a, 602 00 Brno, Czech Republic
xportes@fi.muni.cz

Abstract. We propose an extension to the standard text prompt-based way of interacting with Large Language Models (LLMs), which allows the user to specify a melody (F0 curve) that the generated text should fit. This novel task, which we call Melodic Alignment, involves guiding the text generation process of an arbitrary LLM by the F0 curve, provided by the user as an additional input in the form of an audio recording of either regular speech or a hummed melody. We also propose an ASR-based benchmark to evaluate the performance of various approaches to solving the Melodic Alignment task, along with a discussion on its potential use cases.
Then, as a first-shot model architecture, we propose Melodic Aligner (Meligner). Meligner uses a separate seq2seq model trained on melody / text parallel data to assign a score to each candidate token during the LLM's decoding phase. The 'F0 fit score' each token receives is then merged with the token's probability assigned by the LLM. The degree to which the generated text is influenced by the rescoring can be tweaked by choosing the appropriate merging strategy.
Samples and other supplementary materials will be added to the following link: https://gitlab.fi.muni.cz/nlp/melodic_alignment

Keywords: LLM, large language model, F0, speech melody, speech translation

1 Introduction

Large Language Models (LLMs) have seen a great surge in popularity in recent time, and have been adopted by the general public as useful assistants for a broad variety of tasks. However, for some applications, the text-only mode of interaction with LLMs can be limiting, particularly in cases when the generated text should sound a certain way when read out loud. In this paper, we propose an extension to the standard text prompt framework, which allows the user to specify an audio recording of the melody that the generated text should fit. This recording can either contain actual speech, from which the melody is extracted, or can contain a hummed melody, depending on the use case (discussed in

A. Horák, P. Rychlý, A. Rambousek (eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2023, pp. 133–141, 2023.

section 4). We call this novel task Melodic Alignment and define an ASR-based metric to be used as a performance benchmark.

We outline multiple use cases where the Melodic Alignment framework could be applied, and, as a first-shot approach, we propose Melodic Aligner (Meligner); an F0 alignment scheme using a separate 'rescoring' seq2seq model trained on parallel F0 curve / text data. Meligner extracts the F0 curve from the input recording, converts it into embedding vectors, and subsequently feeds it into the encoder of the rescoring model. During the decoding phase of the LLM, Meligner calculates each candidate token's likelihood that it would have been generated by the rescoring model. This likelihood is then combined with the token's original probability assigned by the LLM, forming a final score based on which the decoding continues. If beam search is used, this happens at every step and in each beam.

2 Related work

There is, to the extent of our knowledge, no system capable of guiding the text generation process of LLMs by speech melody, which is why we are proposing it in this paper. However, there are certian similarities between Melodic Alignment and established NLP tasks, particularly automatic song lyrics translation and automatic speech recognition (ASR).

2.1 Automatic speech recognition (ASR)

Automatic speech recognition can be considered an extreme case of the Melodic Alignment task. If we consider the case when the user specifies only the audio recording, with no textual prompt, the LLM will generate text guided only by the F0 curve extracted from the user-provided recording. This case can be thought of as a constrained version of the ASR task, where only the F0 curve of the audio is used as input. In fact, in our proposed evaluation method, we make use of precisely this task, as described in section 3.2.

2.2 Automatic song lyrics translation

Probably the closest established task to Melodic Alignment is the task of translating song lyrics into tonal languages. In tonal languages, such as Mandarin, multiple words are pronounced in the same way, and speakers differentiate between them based on their pitch. Therefore, the words in the translated lyrics have to match the melody of the song to avoid misunderstandings [2].

There are multiple different approaches to automatic song lyrics generation, including into tonal languages, such as [7], [5] or [6]. Probably most similar to Meligner is the pitch alignment-based rescoring used in GagaST [2]. GagaST uses rescoring during the decoding process to guide the generation of song lyrics conditioned on the song melody. It calculates a pitch alignment score

based on the relationship between the song melody and a pre-determined word tone (one of the 4 tones of Mandarin).

Meligner can be seen as a generalization of the pitch alignment score to arbitrary F0 shapes, where instead of four predetermined tones for each word, the tone of each word is determined dynamically by the rescoring model, taking into account previous context as well.

3 Task definition & evaluation

3.1 Task definition

The goal is to rescore the outputs of an LLM during its decoding stage, in order to make the resulting text fit a user-specified melody (F0 curve). The F0 curve can either be can be extracted from a user-provided audio recording, or provided by the user as a series of frequency values in Hz at a defined sampling rate.

3.2 Task evaluation

As an evaluative benchmark, we propose a scheme utilizing real-world speech / transcription pairs as reference. It can be thought of as a constrained version of the Automatic Speech Recognition (ASR) task, in which only the F0 curve is used from the input speech audio. The evaluative scheme calculates a single benchmarking metric, the average rank-shift score. The average rank-shift score is calculated in the following way.

Given a speech / transcription pair, we provide the speech recording as the audio input and let the LLM generate the first token without any textual prompt. At each decoding step, we define the 'correct' token as the token present in the transcription at the position corresponding to the current decoding step. We then observe whether the rank of the 'correct' token moves up or down after rescoring takes place. A well-performing rescoring scheme should naturally see the words from the transcription climbing up in rank after being rescored. Before moving to the next decoding step, we fix the correct token at the current decoding position to be used as context for the next decoding step.

We define rank-shift as the difference between the 'correct' token's original rank and its rank after rescoring. At the end of the decoding process, the rank-shifts from each step are averaged, and a single average rank-shift value is reported.

4 Use cases

The fact that the Melodic Alignment framework preserves the standard textual prompt of the LLM grants it broad applicability. Any prompt that could benefit from the ability to specify a desired melody the generated text should fit can be considered a use case of Melodic Alignment. In this section, we propose just a few of the possible use cases.

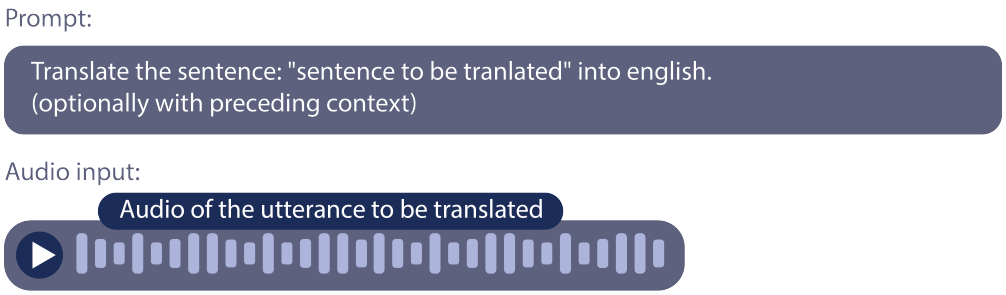

Fig. 1: Example prompt and audio input for the emotional speech translation task.

Emotional speech translation (dubbing) Currently, there are two main approaches to speech translation. There is the traditional cascade approach, where the speech is first converted into text, translated within a text-to-text framework, and then resynthesized back into audio. By using the intermediate textual representation, the model has access to the vast text-to-text translation data, and can therefore deliver high-quality translations, however, at the cost of losing the prosodic information of the original utterance [1].

The other approach, end-to-end speech translation, where parallel audio recordings are used to translate directly from source speech to target speech, is able to preserve the original prosodic content, however suffers from data scarcity, since parallel speech-to-speech data are much less numerous than their text-to-text counterpart [1].

By casting the translation problem into the Melodic Alignment framework, it is possible to obtain the best of both worlds. One could use the cascading approach, thus exploiting the vast amount of text-to-text data, while at the same time have the translation process be driven by the F0 of the original speech, thus preserving the prosodic content of the original.

Fig. 2: Example prompt and audio input for the ASR task.

Automatic Speech Recognition (ASR) in highly noisy settings We hypothesize that when conducting Automatic Speech Recognition under conditions where phonemes are unintelligible, i.e. muffled speech, our approach could lead to generating text that, while not exactly accurate, preserves the intent of the speaker. Whether our approach generates useful transcriptions is up to experimentation.

Copywriting aid As a slightly exotic use case, being able to guide text generation by melody could allow authors to take existing dialogues and write alternative ones with identical emotional content. This could allow one to take, for example, a dialogue about topic A and turn it into a dialogue about topic B while keeping the overall style and emotional impression of the original.

Additionally, when writing, for example, advertisement slogans, Melodic Alignment could make it possible to automatically generate a fitting text to a custom, pre-crafted melody. This could allow copywriters to focus their efforts on crafting the ideal, catchy melody for their slogans, leaving it to the Melodic Alignment model to fill in the words after the fact.

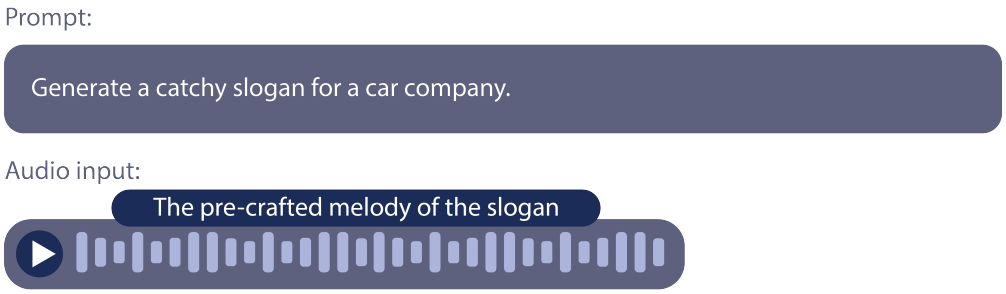

Fig. 3: Example prompt and audio input for the copywriting aid task.

Automatic song translation As mentioned in section 2.2, Melodic Alignment shares many similarities with the task of translating songs into tonal languages. Naturally, a well-performing Melodic Alignment model could be used to solve this task, however, this use case is left for future work, as it will probably require introducing information on the rhythm and other musical features of the original song.

5 Proposed Method: Meligner

As a first-shot architecture, we propose the Meligner scheme. Meligner employs a separate rescoring model; a seq2seq model trained on F0 curve / transcription pairs. The textual prompt is processed in the standard way by the LLM, while

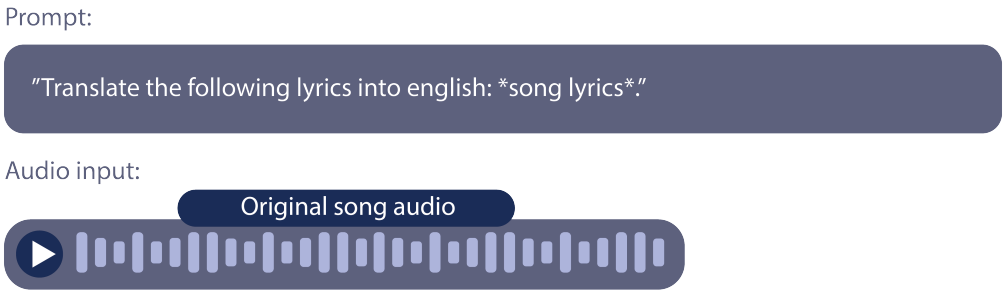

Fig. 4: Example prompt and audio input for the automatic song translation task.

the audio input is first preprocessed, and then fed into the rescoring model's encoder. The rescoring model is then used during the decoding stage of the LLM, where it assigns each candidate token a score representing the 'goodness of fit' between the token and the F0 curve. In the following, we begin by a description of the F0 preprocessing stage, then we discuss the rescoring model and, finally, the rescoring process itself. For an overview of the entire scheme, please refer to figure 5.

5.1 F0 extraction & preprocessing

We extract the F0 curve from the input recording using the YAAPT algorithm [3]. Then, since state-of-the-art seq2seq models expect their input to be in the form of embedding vectors, we encode the F0 curve obtained from YAAPT into a series of vectors by the use of a vector quantized variational autoencoder (VQ-VAE).

F0 extraction The YAAPT algorithm outputs a series of integer values from 0 to 400, which represent the F0 values in Hz, sampled at a user-defined sampling rate from the audio recording.

$$F0\_curve = m_0, m_1, m_2, \ldots, m_{N-1} \in [0, 400]$$

where N = audio length / sampling rate The sampling rate is a hyperparameter whose value is up to experimentation.

F0 embedding To convert the series of integers into vector embeddings, we make use of a vector quantized variational autoencoder (VQ-VAE), which is an approach adapted from the paper [4]. The VQ-VAE framework consists of a convolutional encoder, a bottleneck layer, and a decoder. It is trained using the standard autoencoder objective, i.e. it is trained to reconstruct its own input, under the limited resources constraint imposed by the bottleneck.

To obtain a vector representation of the F0 curve, we feed it into the encoder of the VQ-VAE and extract the latent vectors.

$$EF_0(m) = (h_1, \ldots, h_{L'}), h_i \in \mathbb{R}^{128}$$

Additionally, the latent vectors are each coerced into one of N codebook vectors, with the codebook size N being a tunable hyperparameter. while, in the original paper, only the indices into the codebook are used for further processing, we use the actual codebook vectors.

5.2 Rescoring model

The F0 embedding vectors obtained from the variational autoencoder are fed into the encoder of the rescoring model. The rescoring model is a seq2seq model trained on F0 embedding / text pairs, which serves as a judge of F0 curve / token fit. To teach the model the correspondence between F0 and tokens, we cast the problem into a translation task. We consider the melody embeddings to be a sort of language, and the task of finding words fitting to a given melody is transformed into the task of translating from this 'melody embedding language' to English (or any other language). While any seq2seq model can be used, we use the standard transformer architecture conforming to [8].

5.3 LLM outputs rescoring

The rescoring of LLM outputs during decoding is conducted as follows. At each step of the beam search, and in each of the beams, the rescoring model is used to calculate a score for the top N candidate tokens. This 'F0 fit score' is obtained as the likelihood that the token would be generated by the seq2seq model, given the input melody and the preceding text so far generated by the LLM. Each token's probability assigned by the LLM is then merged with its F0 fit score, and decoding is continued using these new scores.

Tokenization Meligner allows any tokenization to be used, provided that it is shared by both the LLM and the rescoring model. If different tokenization is used for each model, the two tokenization schemes should be compatible in the sense that it should be possible to break down the LLM token into multiple rescoring model tokens. The rescoring model tokens can then be fed sequentially into the rescoring model, with the likelihoods at each step aggregated by multiplying and then taking the n-th root. This is the approach we use in our experiments, as we are using phonemes as our tokens of choice for the rescoring model. We believe that it is at this level of granularity that the relationship between melody and text is the most prominent.

6 Training of models used in Meligner

In order to train the VQ-VAE and the rescoring model, we use a speech / transcription dataset. We split the dataset into two parts, one for training of the VQ-VAE encoder, and the other to train the rescoring model. For an overview of the entire training process, please refer to figure 6.

6.1 F0 encoder (VQ-VAE)

The first split is used to train the F0 encoder. We train the F0 encoder in the same way as [4], i.e. we extract F0 from each recording in the dataset by applying the YAAPT algorithm, and subsequently use it to train the autoencoder, using a reconstruction objective. The transcriptions in the dataset are not used.

6.2 Rescoring model

To train the rescoring model, the other split of the speech / transcription dataset is used. We first convert the dataset into an F0 curve / transcription dataset by extracting the F0 curve from each speech recording using the YAAPT algorithm. Then, we use the F0 encoder we have trained in the previous step to convert the raw F0 curve into a series of vector embeddings. The model is then trained on F0 embeddings/transcription pairs in a standard seq2seq framework.

7 Conclusion

We have proposed Melodic Alignment, a novel task that involves using speech melody, provided as an audio input in addition to the standard textual prompt, to guide the text generation process of Large Language Models (LLMs) We also propose a method for evaluation of performance at the Melodic Alignment task. Since the task is designed to preserve the standard textual prompt of the LLM, its applications are as varied as the number of prompts one can come up with.

Additionally, we have presented a rescoring scheme, Meligner, as a first-shot attempt at solving Melodic Alignment. The scheme is model agnostic and, as such, is compatible with any model architecture. As a next step, we plan to implement the Meligner scheme and evaluate its performance on real-world data, using the proposed evaluation scheme. In case of favorable results, we will apply Meligner to the use cases described in section 4, as well as investigate applying the same approach to other prosodic features, such as stress or rhythm.

References

1. Bentivogli, L., Cettolo, M., Gaido, M., Karakanta, A., Martinelli, A., Negri, M., Turchi, M.: Cascade versus direct speech translation: Do the differences still make a difference? In: Annual Meeting of the Association for Computational Linguistics (2021), https://api.semanticscholar.org/CorpusID:235293674

2. Guo, F., Zhang, C., Zhang, Z., He, Q., Zhang, K., Xie, J., Boyd-Graber, J.: Automatic song translation for tonal languages. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Findings of the Association for Computational Linguistics: ACL 2022. pp. 729–743. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.findings-acl.60, https://aclanthology.org/2022.findings-acl.60
3. Kasi, K.: Yet another algorithm for pitch tracking (yaapt) by (2002), https://api.semanticscholar.org/CorpusID:15301096
4. Polyak, A., Adi, Y., Copet, J., Kharitonov, E., Lakhotia, K., Hsu, W.N., rahman Mohamed, A., Dupoux, E.: Speech resynthesis from discrete disentangled self-supervised representations. In: Interspeech (2021), https://api.semanticscholar.org/CorpusID:262491522
5. Qian, T., Lou, F., Shi, J., Wu, Y., Guo, S., Yin, X., Jin, Q.: UniLG: A unified structure-aware framework for lyrics generation. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 983–1001. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.acl-long.56, https://aclanthology.org/2023.acl-long.56
6. Qian, T., Shi, J., Guo, S., Wu, P., Jin, Q.: Training strategies for automatic song writing: A unified framework perspective. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4738–4742 (2022). https://doi.org/10.1109/ICASSP43922.2022.9746818
7. Tian, Y., Narayan-Chen, A., Oraby, S., Cervone, A., Tao, C., Sigurdsson, G., Zhao, W., Chung, T., Huang, J., Peng, V.: Unsupervised melody-to-lyric generation. In: ACL 2023 (2023), https://www.amazon.science/publications/unsupervised-melody-to-lyric-generation
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)